GLOBAL COVID MANAGEMENT SYSTEM

ESOF5014: AGILE SOFTWARE DEVELOPMENT

FINAL PROJECT REPORT

Project Supervisor: Dr Yassine Abdulsalam

Contributing Members:

Prateek Asthana – 0515997

Jun Li - 1100472

Kolinben Rakeshkumar Sukhadia- 1159716

Malavi Gheewala - 1111701

Mohan Ram Selvaraj - 1152878

Shravan Kumar Chinthapandu -1132853

VedangNilotpal Yagnik - 1159608

Shahrukh Sattar Chowdhury - 1147198

Srikanth Patamsetti – 1153203

ABSTRACT

Corona virus also known as Covid-19 greatest danger to the world. Over 136 million people are affected, and 2.94 million people died across the world and 77.3 million People recovered. Covid 19 has changed the way we live and work. As numerous regions were affected by the pandemic, several countries faced a sudden 'lockdown'. With communities confined to their own homes, organizations of all sizes, with little or no time to plan, had to alter the way they worked or shut down entirely. In 2020, 8.8 per cent of global working hours were lost relative to the fourth quarter of 2019, equivalent to 255 million full-time jobs. On March 11, 2020, the World Health Organization (WHO) characterized COVID-19 as a pandemic. The World is not prepared for this pandemic, we are still fighting this Corona Virus in our daily life. One of the major problems we are fighting in the pandemic is a Management System. The world lacks a covid management system that not only can work per region but also on a global scale. World needs to come together and create a Covid Management System where people can view and create covid profiles anonymously. Global Problems requires global solution. We propose *Global Covid Management System* a platform that not only can work per region but also on a global scale. We can consolidate worldwide data regarding covid and it is available to the public. We feel by involving the public in the crisis, we can increase the number of people who are fighting to eradicate covid-19. The website has a registration/login page that can contain functions. People can create their profiles which include name, address, phone, job title, current earnings. People in contact and the health care professional can input covid results in the system and the general population can view it. We also have a plan to use the data to create geo data analytical maps to show covid heavy zones, job loss because of covid and current earnings of population based on region.

1. INTRODUCTION

Agile methodologies are currently one of the most well-known and widely used in the software development industry. Its key goal is to keep customers happy by providing consistent service. A project management methodology where a large project is segregated into smaller iterations is known as Agile. Agile is described simply by the Agile Alliance, which was established by the 17 signers of the Agile Manifesto, as "the ability to build and adapt to change in order to succeed in an unpredictable and chaotic world". These members carefully analysed the defects, or the problems faced and came up with new methodology for software development. The Agile Alliance group agreed upon the “Agile Manifesto”, which has four values, and it has 12 principles. Based on these principles and values, developers and software researchers developed different Agile methodologies frameworks such as Scrum, Kanban, Extreme programming, Lean development, Crystal methods, Featured Driven Development, DSDM (Dynamic System Development Method).

We developed the project in Agile approach. We had Regular Scrum meetings, Sprint Planning, Retrospective meetings, and Release Planning. To avoid complexity we used JIRA, Discord, GitHub, Google Docs, and Lucid chart to keep our project simple. We took collective ownership in dividing the stories into tasks, and the team members sign up for them individually. At the end of each iteration, we had an actual working product. At the end of each sprint we had retrospective meeting, to learn from our mistakes.

2. MODELING

The decomposition of work packages and plans is depicted in a workflow diagram in the software development phase. For our project, we followed WFD to the core to achieve optimum efficiency. To produce the product on time, a Global covid management project needs a well-defined WFD. We divided the work into two parts which included agile workflow and a project workflow to make task easier to understand.

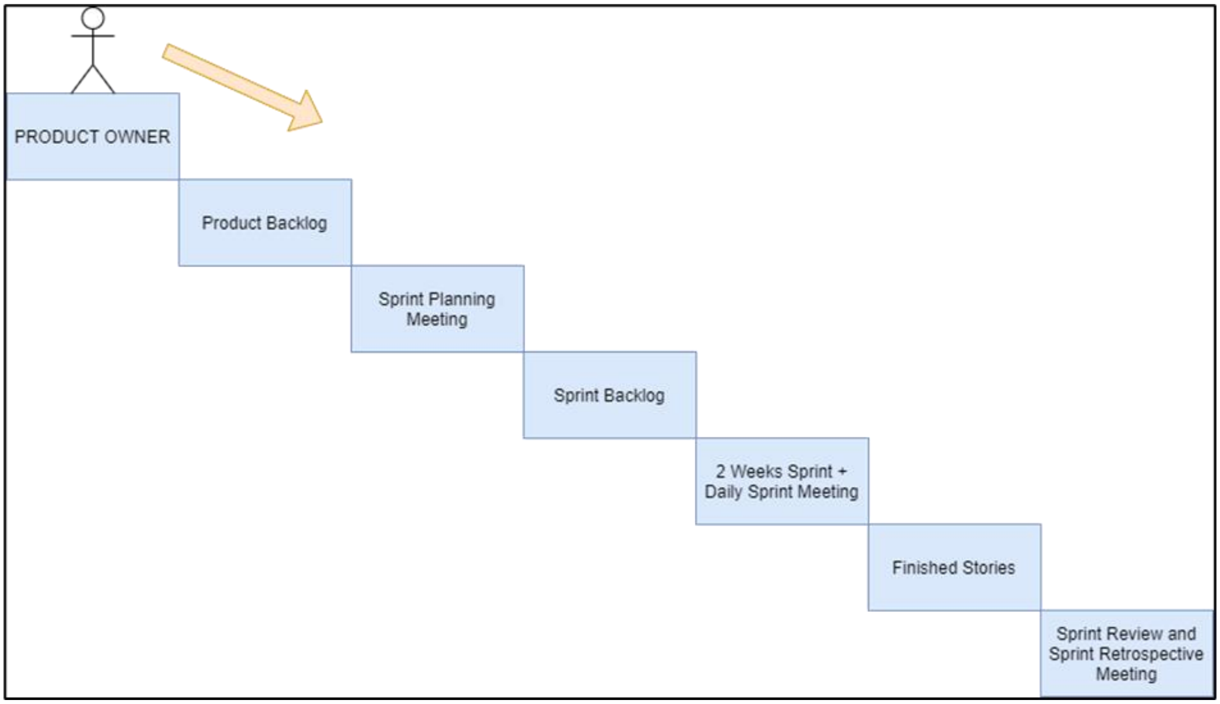

PRODUCT OWNER
Product Backlog
Sprint Planning Meeting
Sprint Backlog
2 Weeks Sprint + Daily Sprint Meeting
Finished Stories
Sprint Review and Sprint Retrospective Meeting

Figure 1 AGILE WORKFLOW

To achieve highest possible efficiency, we stuck to the agile practises. Our team had a daily Stand-up meeting of 15 mins after every lecture for sharing task and progress/done by team members. Initial sprint duration was about 2 weeks and then as we had a clearer idea it was reduced to 10 days. Retrospective meeting was done to improve our working methodologies.

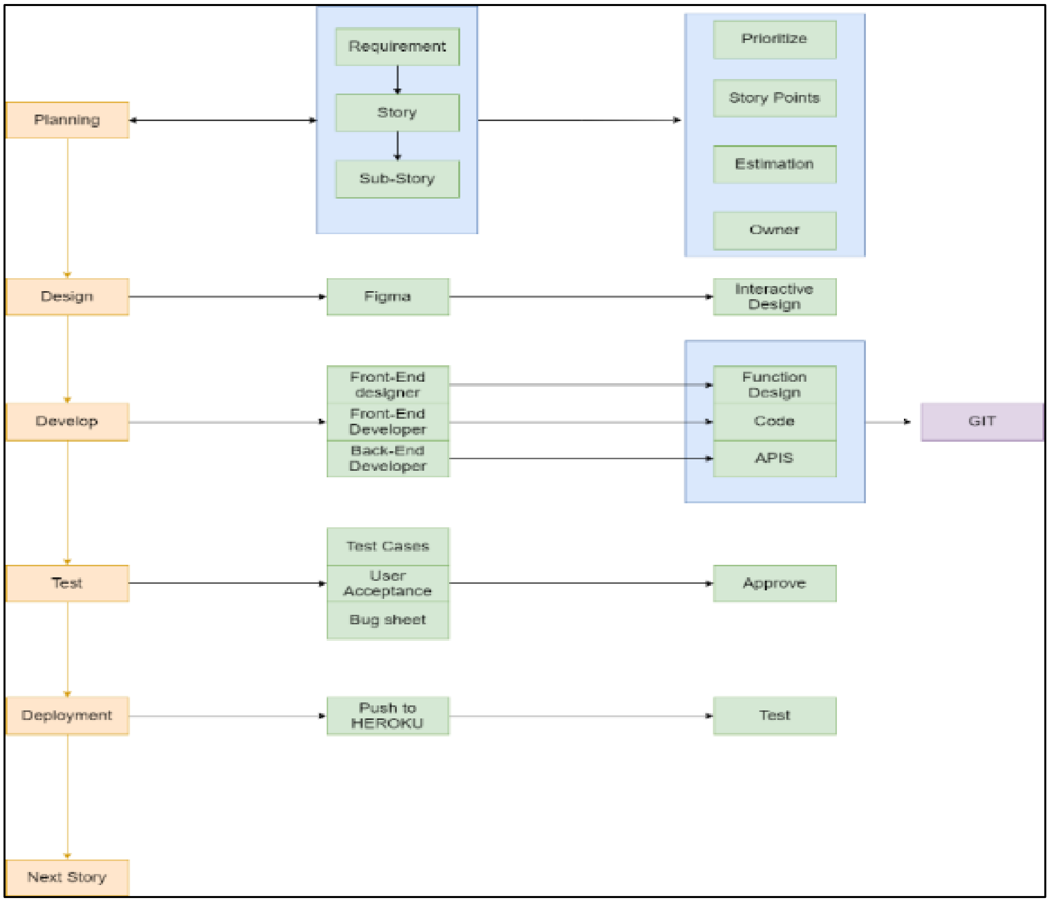


Figure 2 PROJECT WORKFLOW

Agile planning is a project planning method that estimates work using self-contained work units called iterations or sprints. At first, the stakeholders are defined, and the business criteria are specified. Internal and external stakeholder relationships were formed, and meetings were registered. We started dividing our project into phase which could give us a clear idea on how to move forward.

2.1 Phase 1: Research, Literature review and Documentation regarding agile

Considering how Covid-19 has managed to screw up the lives of millions of people and is continuing to grow, we need to study about how it originated, what factors should be considered in building a global covid management system. It is an ongoing study so daily new advancements are made to contain the virus. To identify the gaps and inconsistencies in the research and build a foundation to support our topic it is crucial to study the past work.

After the coronavirus outbreak, in response to this ongoing public health emergency, an online interactive dashboard is developed and hosted by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University, Baltimore, MD, USA, to visualise and track reported cases of coronavirus disease 2019 (COVID-19) in real time. The dashboard illustrates the location and number of confirmed COVID-19 cases, deaths, and recoveries for all affected countries. It was developed to provide researchers, public health authorities, and the public with a user-friendly tool to track the outbreak as it unfolds. All data collected and displayed are made freely available, initially through Google Sheets and now through a GitHub repository, along with the feature layers of the dashboard.

All data collection and processing were done manually, and updates were typically done twice a day, morning, and night (US Eastern Time). As the outbreak evolved, the manual reporting process became unsustainable. To identify new cases, we monitor various Twitter feeds, online news services, and direct communication sent through the dashboard. Before manually updating the dashboard, we confirm the case numbers with regional and local health departments, including the respective centres for disease control and prevention (CDC) of China, Taiwan, and Europe, the Hong Kong Department of Health, the Macau Government, and WHO, as well as city-level and state-level health authorities.

Development of this web application using the agile model makes it easier to work in a cross-functional team and make the development and deployment process much faster. In the initial one or two week we had some design/features ready demonstrating what we are building. We used JIRA as an agile project management tool for managing tasks, checking progress and GitHub repository for managing the code base between the team to work parallelly.

2.2 Phase 2: Project Inception and understanding business case.

As Ux is the totality of end-user’s perceptions as they interact with a service. These perceptions include effectiveness, efficiency, emotional satisfaction, and the quality of the relationship with the entity that created the service. If a user is unable to communicate with your platform efficiently, they will not have a good experience when doing so. So, to make the service more engaging, it should consist of interaction design, user interface, visual design, content, information architecture, functionality, usability, and typography. Service page design was simple in days gone by; designers designed products that they felt were cool and they hoped their customers would enjoy. Sadly, with that strategy, there are two issues. The first is that there was much less competition, back then, for the attention of people online. The second is that in that approach, there is little concern for the customer of the product at all; the success or failure of a production project was down to chance as much as it was down to the design team's judgment. Focusing on UX makes it possible to concentrate on the consumer by design.

2.3 Phase 3: Software Modelling

Modeling is where the web application's design and analysis take place, with the aid of software developed specifically for this purpose. Since written or spoken word communication is not a precise way to represent the functions provided by the web application, modeling is essential. At the beginning of project, the team decided to do some initial, high-level architectural modeling to identify a viable technical strategy for the solution. Using case diagrams, sequence diagrams, architecture, and other modeling techniques we achieved that.

2.3.1: Unified Modeling Language

We used UML to do basic modeling. Two types of UML diagram including use case diagram and sequence diagram are used in our initial draft design. We use traditional three-layer web solutions to implement our system. So, the logic could be easily demonstrated through sequence diagram.

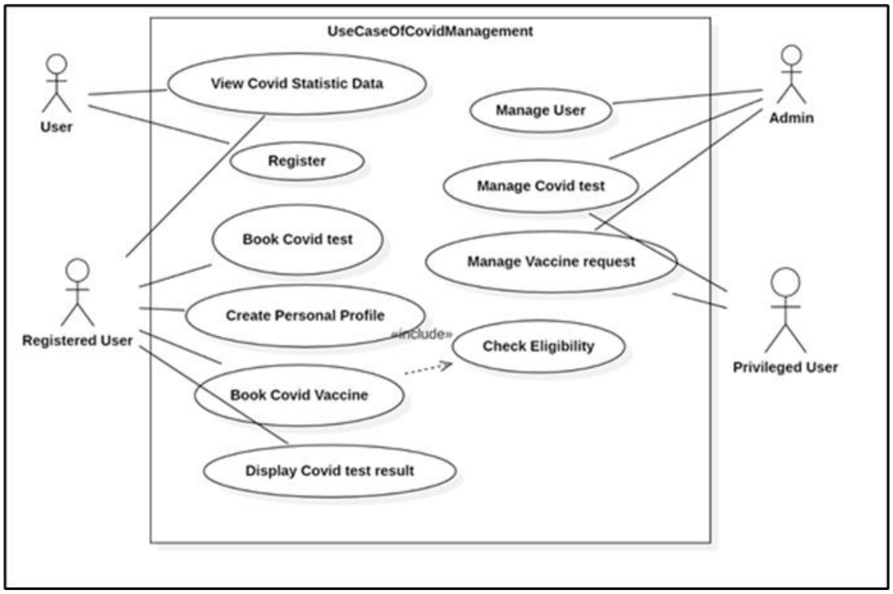


Figure 3 USE CASE DIAGRAM

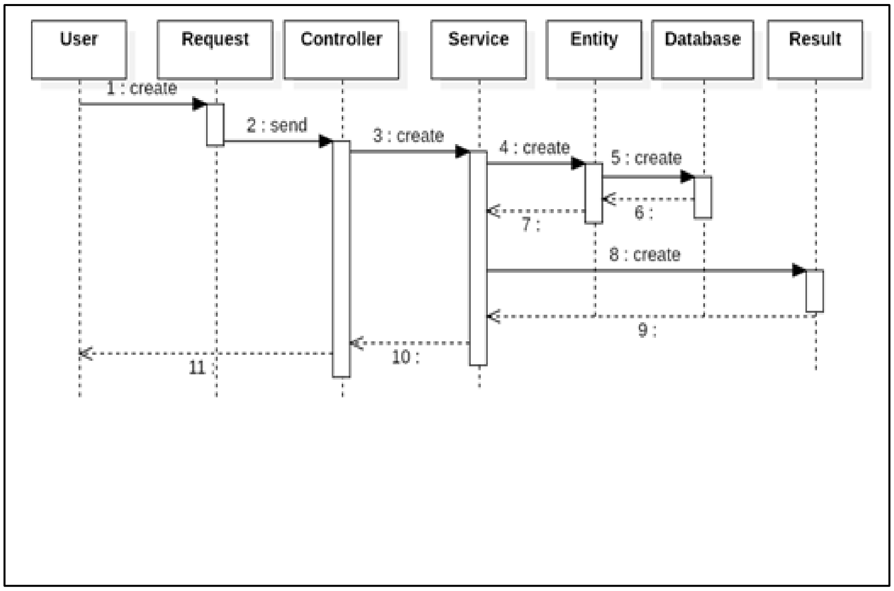


Figure 4 SEQUENCE DIAGRAM

The Controller layer was used to intercept the user's requests and direct the requests to suitable services or pages. The Service layer aimed to provide business logic to handle the user requests.

The Entity layer were classes to hold data which were used in processing and persistence. The database is used to store all data that needs to be persisted. We also use REST API to do frontend-backend communication because frontend and backend could be developed asynchronously. Another advantage is REST API is easy to test. We use SoapUI to do integration tests and regression tests. Although the free version of SoapUI could not support test automation, we could still use it manually by clicking all test items with order because the project size was relatively small, and it still could save a lot of time.

2.3.2: Architecture

In figure 5 below, the 3-tier MERN stack architecture is being used for this project. This makes designing, development and testing more modular and easier as the project is divided into frontend, backend, and cloud database management part. It is a stack which is a collection of various technologies used to create a dynamic web application and It will be made using multiple technologies such as frameworks, libraries, and databases etc. The MERN stack is a JavaScript stack that is used to make the development process very smooth. It is collection of four open-source components: MongoDB, Express, React, and Node.js. All these components provide an end-to-end framework for frontend and backend developers to work in. The main benefit for developers using the MERN stack is that every line of code is written in JavaScript. JavaScript is used everywhere, both for client-side code and server-side code. With one language across tiers, there is no need for context switching. Using the JavaScript stack, developers only need to have understood in JavaScript and JSON. Overall, The MERN stack enables programmers to develop highly efficient web applications. We have frontend which uses Json Web Token (JWT) for authentication and all the RESTAPIs end points after login use that token to verify authenticated user and signature.

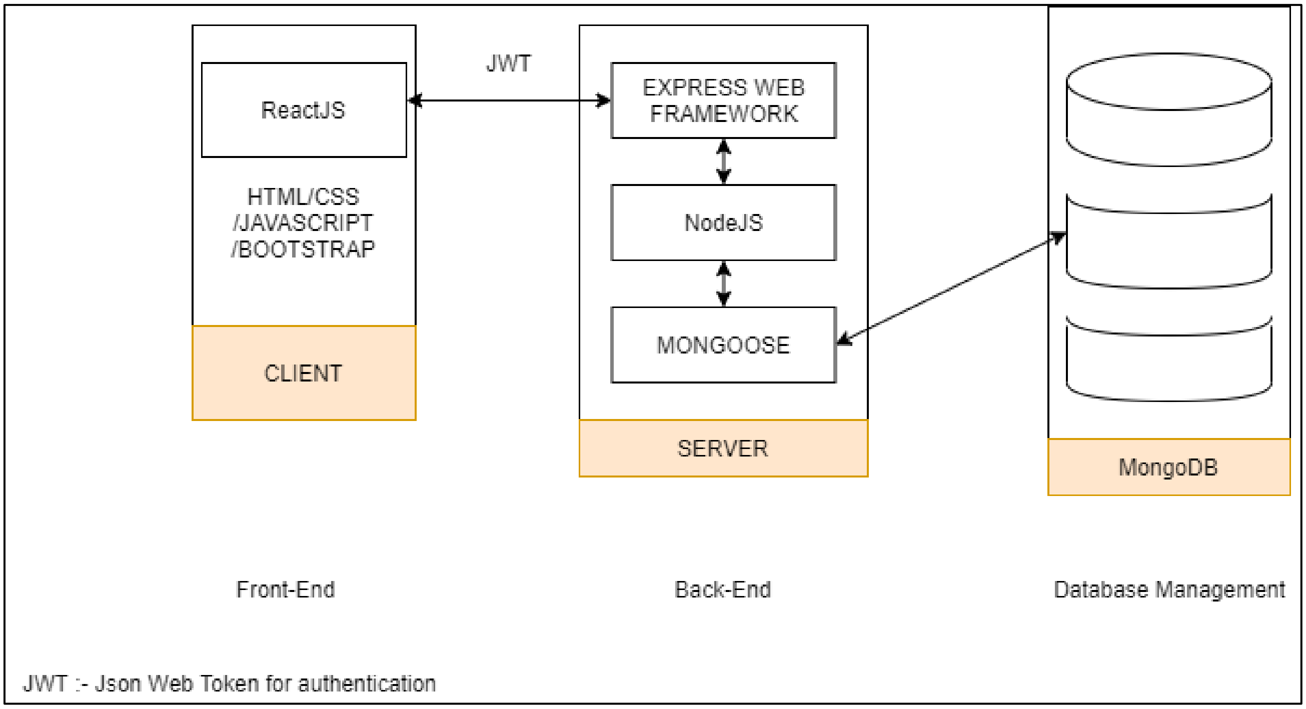


Figure 5 3-TIER ARCHITECTURE

Figure 6 depicts the internal architecture of our web application. There are four web pages Home, News, About and Contact that are public pages which does not require user authentication and those pages can be viewed or traversed through links in the navigation bar without any restriction. Home page is our introduction part where we show information for how to fight covid pandemic and precautions and measures that should be taken to prevent from getting it. News page fetches news from verified public sources online using News APIs and fetch data related to covid that gives instant insight to all users about what is trending. About and Contact pages are for describing who are we, what we do and what is our project about.

There are REST API end points such as login, register and patient with CRUD operations. We have used access control list (ACL) to maintain separate roles that is Admin and User. Admin can be anyone such as government body, public health sector, private or public hospitals who is going to use our web application as their portal. Admin can sign in and can handle all of their patient data and update their covid result directly into the system. On the other hand, User can login to book covid test by filling out form for them as well as their family members and can directly see result of covid test inside the dashboard. All of end points are connected via routes from frontend to backend and backend communicates with cloud MongoDB directly to handle all the data in database.

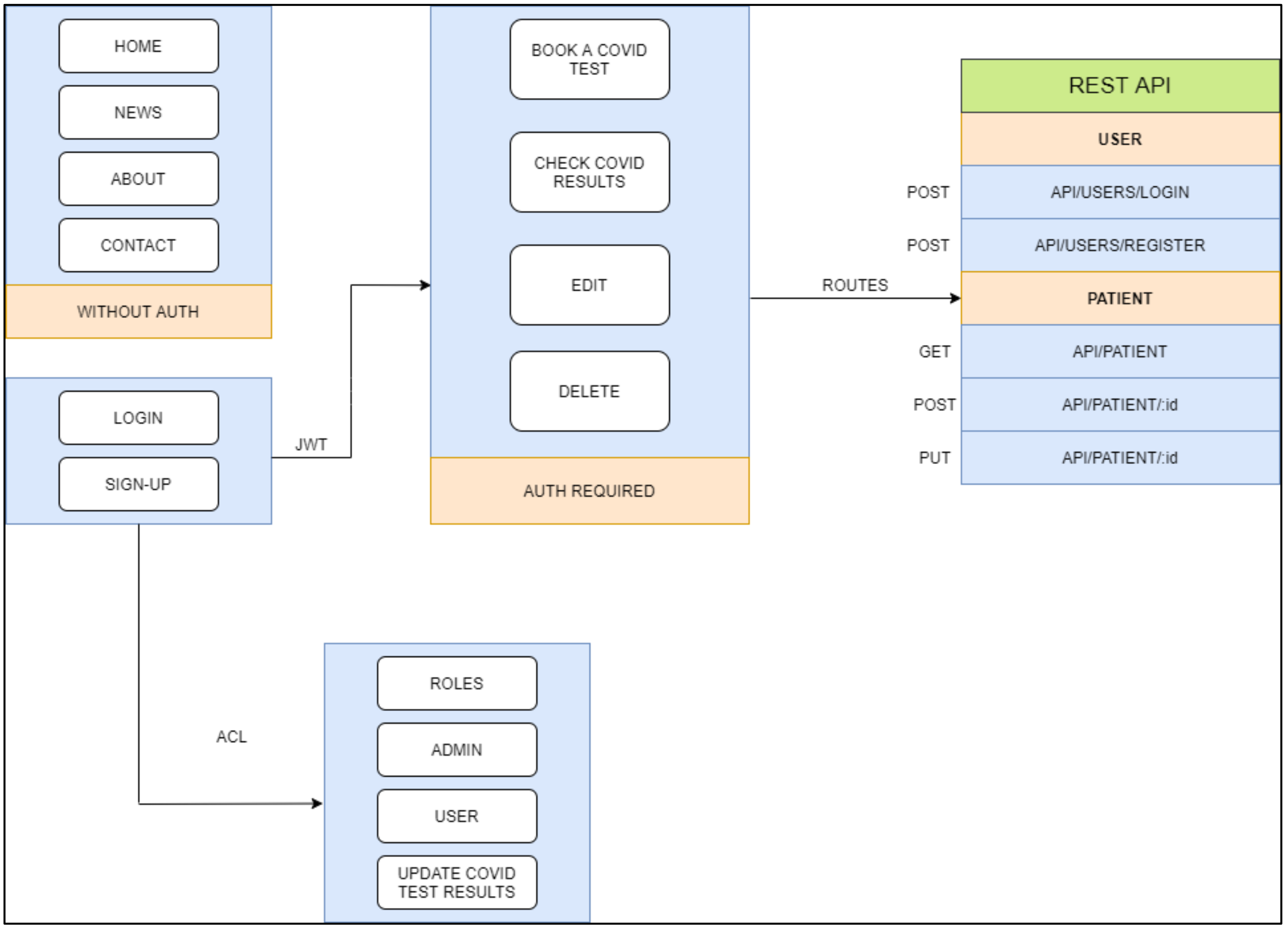


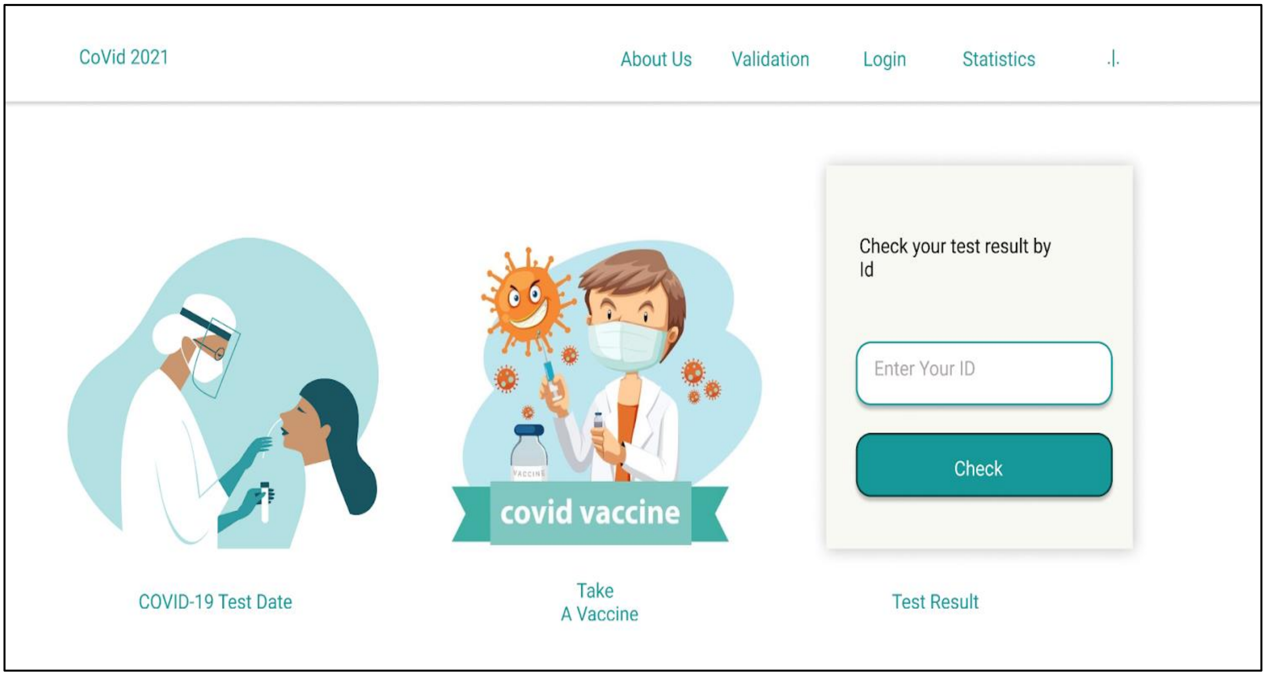
Figure 6 INTERNAL ARCHITECTURE

2.4 Phase 4: Construction

2.4.1 User Experience

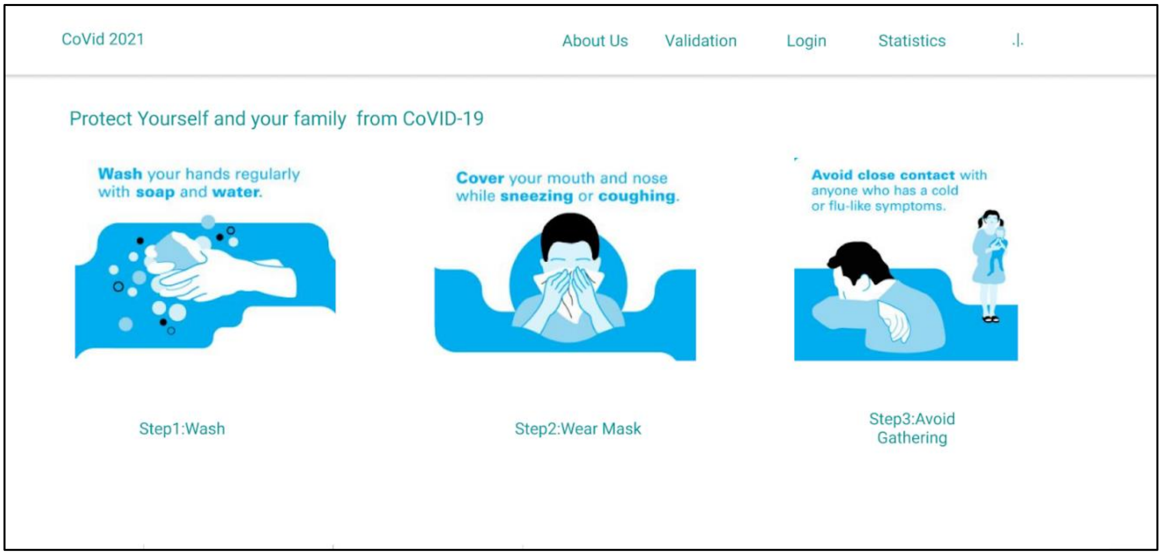
User Experience is the totality of end-user's perceptions as they interact with a service. These perceptions include effectiveness, efficiency, emotional satisfaction, and the quality of the relationship with the entity that created the service. If a user is unable to communicate with your platform efficiently, they will not have a good experience when doing so. So, to make the service more engaging, it should consist of interaction design, user interface, visual design, content, information architecture, functionality, usability, and typography. Service page design was simple in days gone by; designers designed products that they felt were cool and they hoped their customers would enjoy. Sadly, with that strategy, there are two issues. The first is that there was much less competition, back then, for the attention of people online. The second is that in that approach, there is little concern for the customer of the product at all; the success or failure of a production project was down to chance as much as it was down to the design team's judgment. Focusing on UX made it possible to concentrate on the consumer by design. Finally, by empathizing and testing with the testers in our group we came to a very simple and efficient design for a prototype.

For this project, at first, a hand sketch was made by research and empathizing about the thoughts of our testers. Some tools were used to find a bit more information about service users. Many scenarios as possible was defined and considered for the consequences of the service goals. Then, it was followed by the ideate part where we all the data was put together to brainstorm some ideas. This was in the form of storyboards. After the ideate part, the most productive way to give life to prototyping/wireframing for the production was built.



Check your test result by Id

Figure 7 LOGIN PAGE



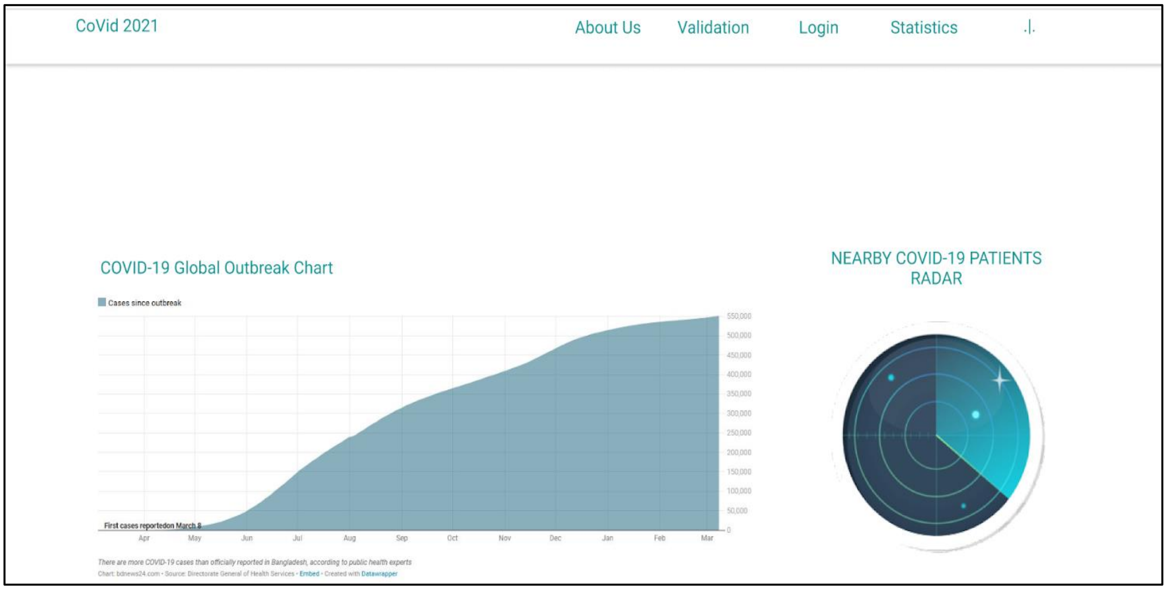
Protect Yourself and your family from CoVID-19

Wash your hands regularly with **soap** and **water**.

Cover your mouth and nose while **sneezing** or **coughing**.

Avoid close contact with anyone who has a cold or flu-like symptoms.

Figure 8 GUIDELINE PAGE



There are more COVID-19 cases than officially reported in Bangladesh, according to public health experts

Chart: bdnews24.com • Source: Directorate General of Health Services • Embed • Created with Datawrapper

Figure 9 GEO-STATISTICAL PAGE

With the time and money available, the project is almost complete, and the developer has refactored the code several times so that new functionality can be easily implemented in the future. Simply to make the program user-friendly, the aesthetic design was held. The application's first few pages are included below.

2.4.1 Home Page

This is homepage design of our global covid management system. From this home page, user can go to News page, about page, Contact page. These are publicly accessible.

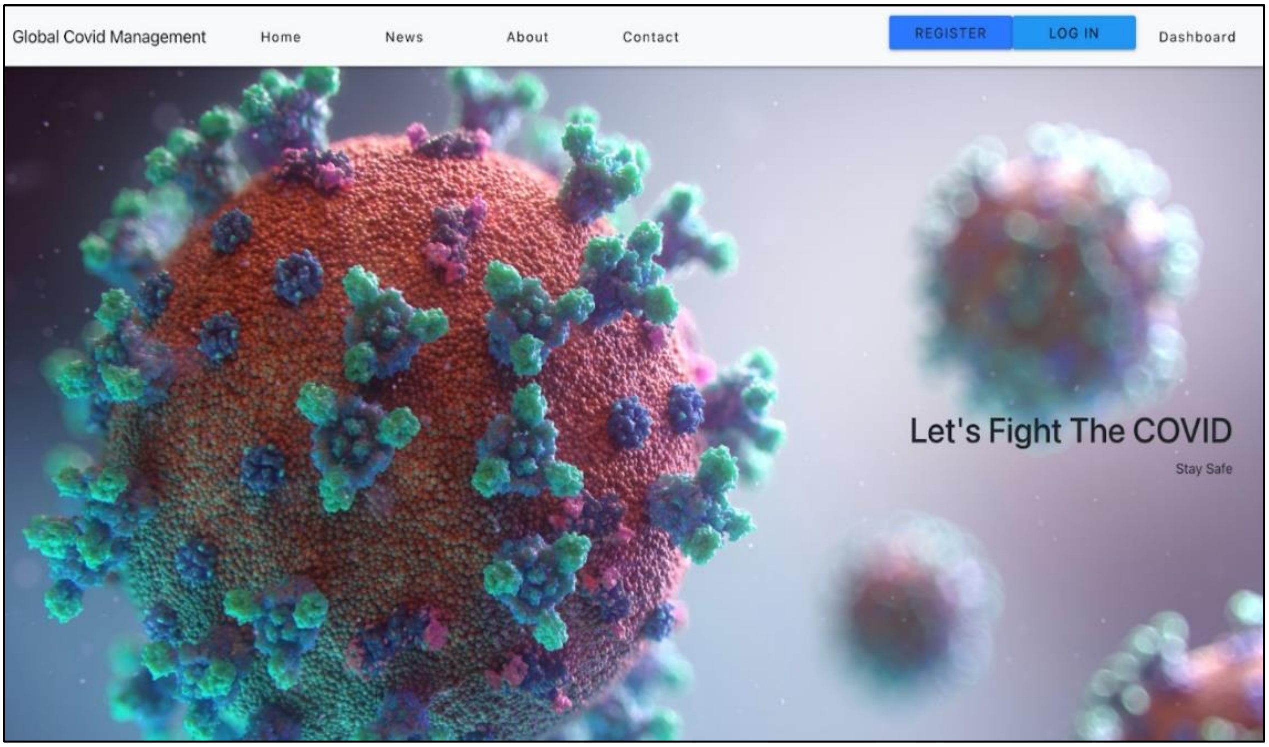
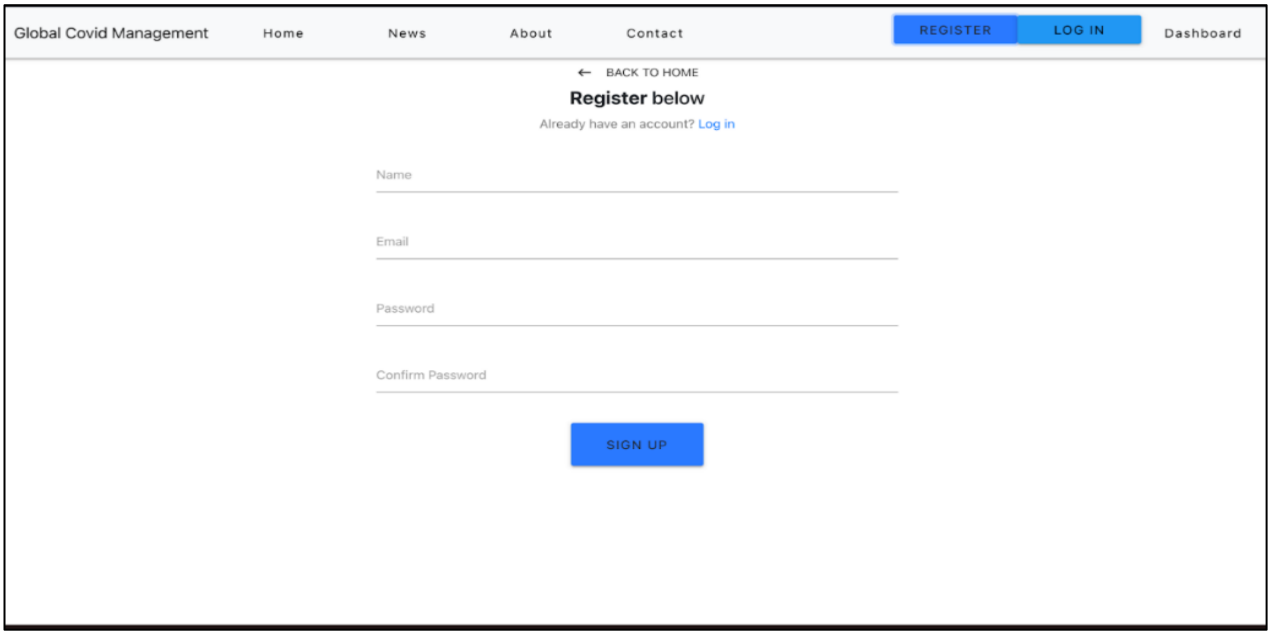


Figure 10 HOME PAGE

2.4.2 Registration Page

Registration page of global covid management system is as below. For booking a covid test, user must do registration from registration page. For registration user must enter name, email and password. After signup user can login.



Register below

Already have an account? Log in

Name

Email

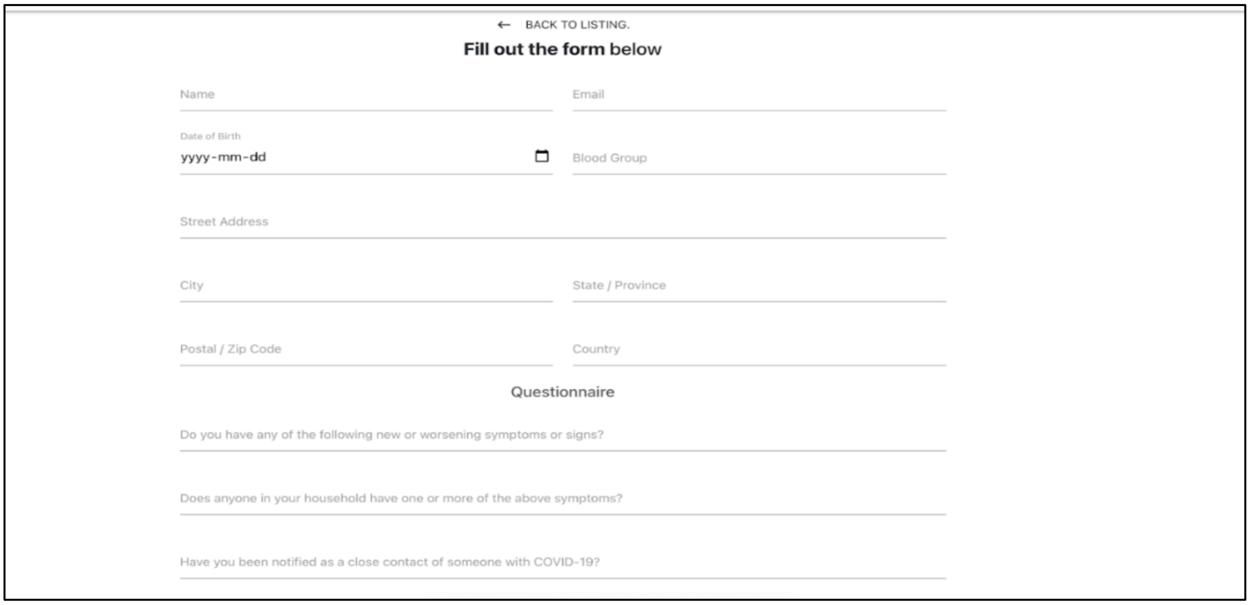
Password

Confirm Password

Figure 11 REGISTRATION PAGE

2.4.3 Booking a covid test.

This is page for filling out form for booking a covid test. Any user who registers and login can book a covid test. For booking a covid test, user must enter name, email, date of birth, street address, city, state, zip code and country.



Fill out the form below

Name

Email

Date of Birth

yyyy-mm-dd

Blood Group

Street Address

City

State / Province

Postal / Zip Code

Country

Questionnaire

Do you have any of the following new or worsening symptoms or signs?

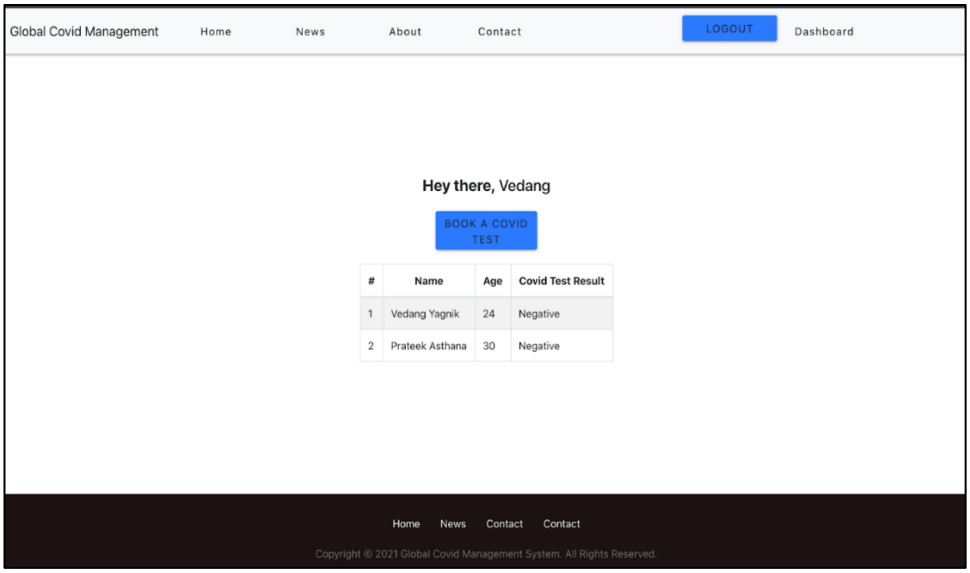
Does anyone in your household have one or more of the above symptoms?

Have you been notified as a close contact of someone with COVID-19?

Figure 12 QUESTAIONAIRRE PAGE

2.4.4 Result Page

This page is showing result that whether person's covid test is positive or negative on result page, there is age calculated based on date of birth.



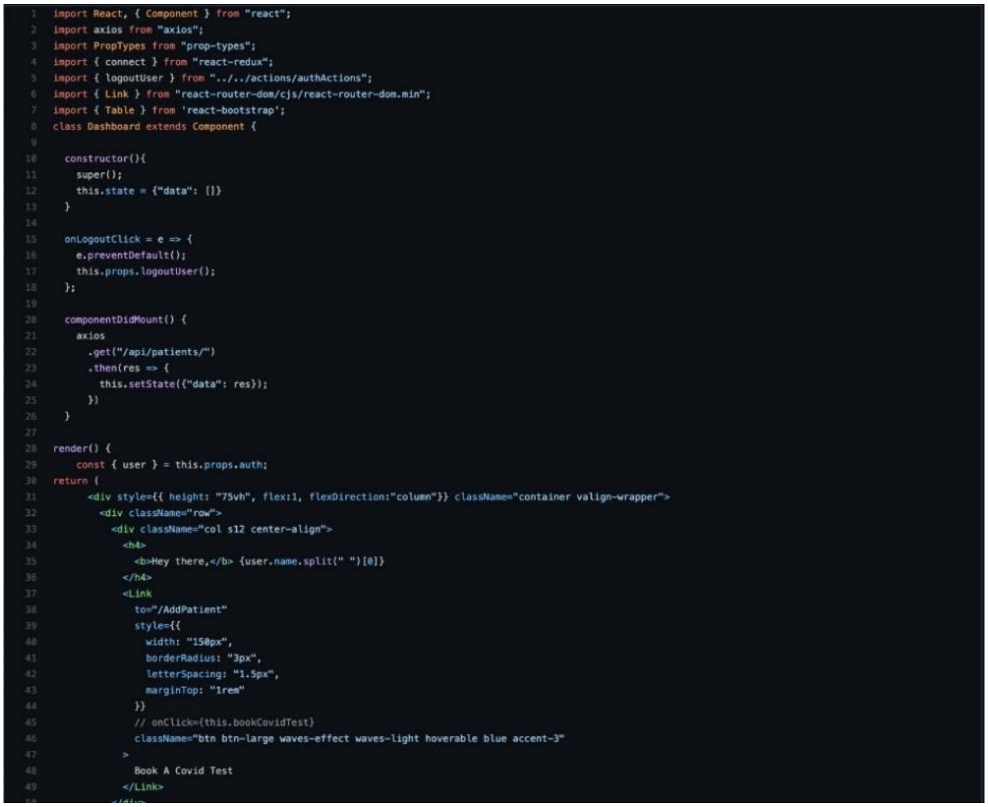
Hey there, Vedang

#	Name	Age	Covid Test Result
1	Vedang Yagnik	24	Negative
2	Prateek Asthana	30	Negative

Copyright © 2021 Global Covid Management System. All Rights Reserved.

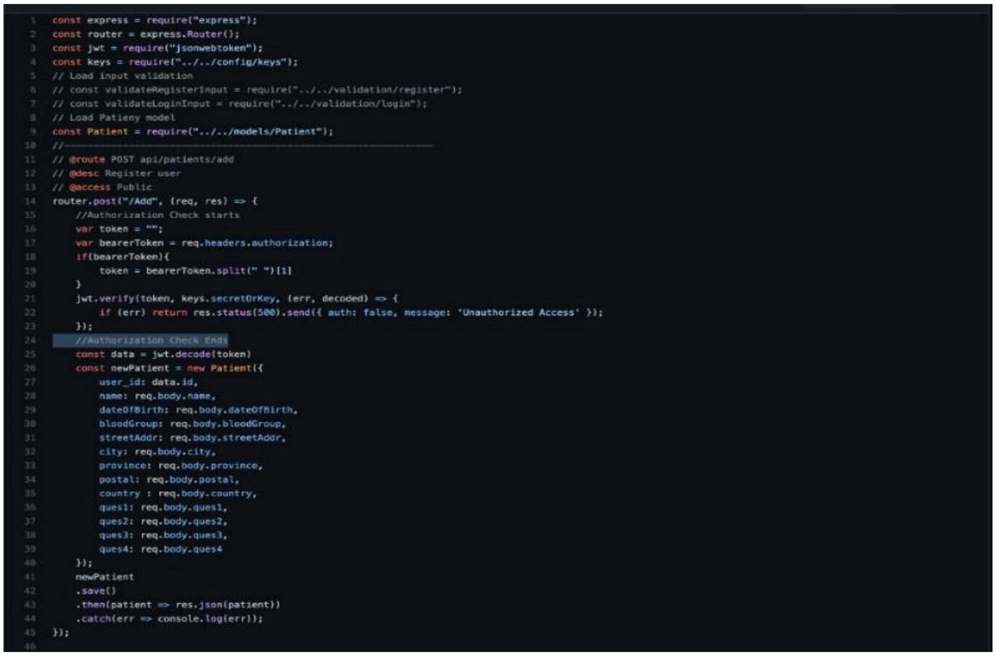
Figure 13 RESULT PAGE

2.5 Phase 5: Deployment



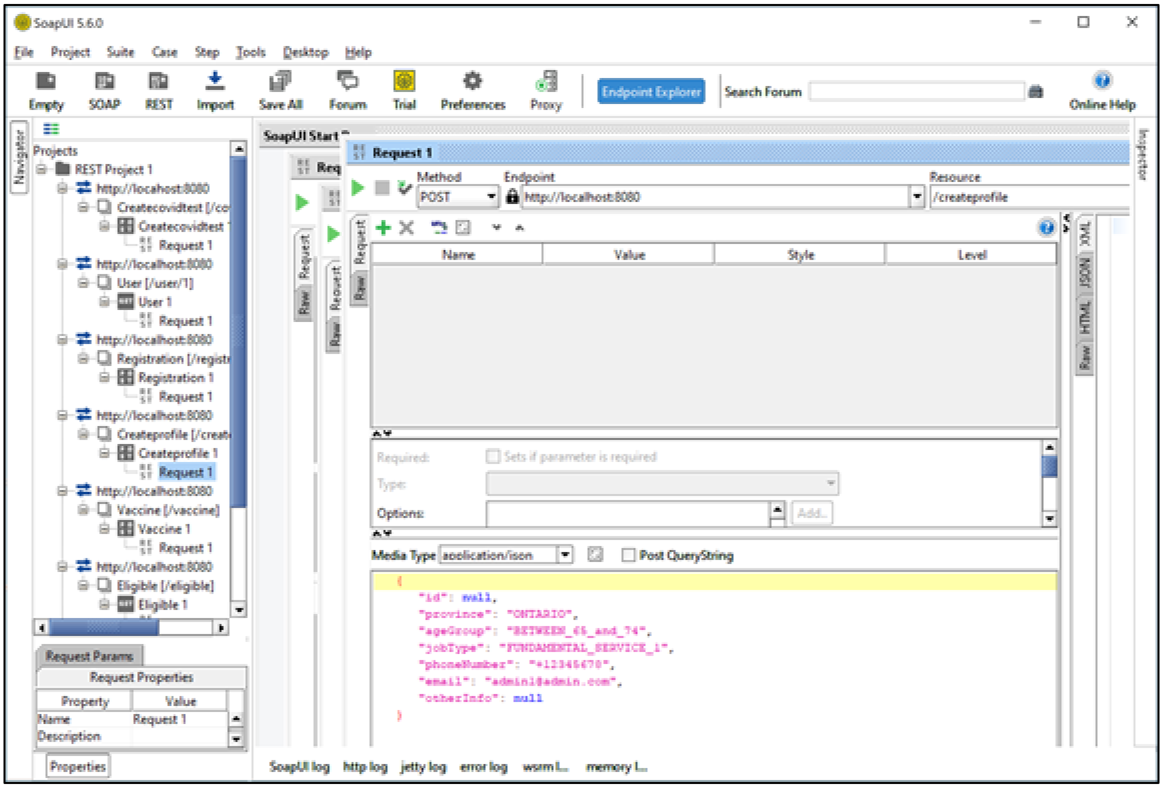
```
import React, { Component } from "react";
import axios from "axios";
import PropTypes from "prop-types";
import { connect } from "react-redux";
import { logoutUser } from "../../actions/authActions";
import { Link } from "react-router-dom/cjs/react-router-dom.min";
import { Table } from 'react-bootstrap';
class Dashboard extends Component {

  constructor(){
    super();
    this.state = {"data": []}
  }

  onLogoutClick = e => {
    e.preventDefault();
    this.props.logoutUser();
  };

  componentDidMount() {
    axios
      .get("/api/patients/")
      .then(res => {
        this.setState({"data": res});
      })
  }

render() {
    const { user } = this.props.auth;
return (
      <div style={{ height: "75vh", flex:1, flexDirection:"column"}} className="container valign-wrapper">
        <div className="row">
          <div className="col s12 center-align">
            <h4>
              <b>Hey there,</b> {user.name.split(" ")[0]}
            </h4>
            <Link
              to="/AddPatient"
              style={{
                width: "150px",
                borderRadius: "3px",
                letterSpacing: "1.5px",
                marginTop: "1rem"
              }}
              // onClick={this.bookCovidTest}
              className="btn btn-large waves-effect waves-light hoverable blue accent-3"
            >
              Book A Covid Test
            </Link>
```

Figure 14 FRONT-END UI



```
const express = require("express");
const router = express.Router();
const jwt = require("jsonwebtoken");
const keys = require("../../config/keys");
// Load input validation
// const validateRegisterInput = require("../../validation/register");
// const validateLoginInput = require("../../validation/login");
// Load Patieny model
const Patient = require("../../models/Patient");
//--------------------------------------------------------------------
// @route POST api/patients/add
// @desc Register user
// @access Public
router.post("/Add", (req, res) => {
    //Authorization Check starts
    var token = "";
    var bearerToken = req.headers.authorization;
    if(bearerToken){
        token = bearerToken.split(" ")[1]
    }
    jwt.verify(token, keys.secretOrKey, (err, decoded) => {
        if (err) return res.status(500).send({ auth: false, message: 'Unauthorized Access' });
    });
    //Authorization Check Ends
    const data = jwt.decode(token)
    const newPatient = new Patient({
        user_id: data.id,
        name: req.body.name,
        dateOfBirth: req.body.dateOfBirth,
        bloodGroup: req.body.bloodGroup,
        streetAddr: req.body.streetAddr,
        city: req.body.city,
        province: req.body.province,
        postal: req.body.postal,
        country : req.body.country,
        ques1: req.body.ques1,
        ques2: req.body.ques2,
        ques3: req.body.ques3,
        ques4: req.body.ques4
    });
    newPatient
    .save()
    .then(patient => res.json(patient))
    .catch(err => console.log(err));
});
```

Figure 15 BACK-END REST-API



Method: POST

Endpoint: http://localhost:8080

Resource: /createprofile

Media Type: application/json

□ Post QueryString

```
{
    "id": null,
    "province": "ONTARIO",
    "ageGroup": "BETWEEN_65_and_74",
    "jobType": "FUNDAMENTAL_SERVICE_1",
    "phoneNumber": "+12345678",
    "email": "admin1@admin.com",
    "otherInfo": null
}
```

Request Properties

Property	Value
Name	Request 1
Description	

Figure 16 SoapUI

3.PROJECT ORGANIZATION CHART

- Jun Li - Back-End Developer – He maintained and developed the backend code necessary for achieving functionalities and satisfying the criteria of the API request.
- Kolinben Rakesh Kumar Sukhadia- Front-End Designer – She is responsible in developing interactive UI and web pages. She also ensures whether proper functionalities are achieved in the front-end.
- Malavi Gheewala - Business Analyst – She is responsible for the communication channel between the developers, suggestion for tasks added in sprints and organize the team for arranging retrospectives and generating essential documentation for this project.
- Mohan Ram Selvaraj - Back-End Developer – He maintains the databases and writes code for fetching/dumping data into the DB. He develops database communication codes and related functionalities.
- Prateek Asthana - Product Owner/ Scrum Master– Being a product owner and a scrum master he helps in maintaining product backlogs, solving impediments and help team to stay focused on agile process.

- Shravan Kumar Chinthapandu - QA/Tester – He is responsible to maintain the quality of product and run test case for ensuring functionalities.
- Shahrukh Sattar Chowdhury - UI/UX Designer – Being the designer he helps to design web pages and help maintain the consistency across web pages. Also, been responsible to improve the aesthetics of the UI.
- Srikanth Pattamsetti – QA/Tester – He used to analyse the use cases and test the product quality for achieving the end goal. He also does the REST API testing and runs regression test.
- Vedang Nilotpal Yagnik - Full-Stack Developer – He develops and maintains the server communications and integration between backend and frontend. He also helped in the frontend development and creating web pages.

4.TOOLS USED

JIRA - JIRA being a common type of agile development tool which helps in prioritizing and planning the tasks, it is going to be utilized here. Moreover, the flexibility of JIRA allows the team to plan, evaluate the project with Scrum and Kanban methods. It has features that can be added to backlogs, bug fixing, story points, sprint reports and burndown charts are some of the tools for tracking the progression of the project.

GitHub: It is a code hosting platform mainly for collaboration and version control. It allows many people to work on the same project simultaneously from anywhere. There are built in tools which enable the user to review the code which is an essential part of the maintaining quality of project. For a team of nine members with a cross-functioning platform GitHub is the best option to have a common workplace and especially for an agile project. It helps to have a proper track on the code. It also helps us revert to older versions in case of any breakages. Providing a great hand on maintenance part of the project.

Discord**:** Proper communication and continues knowledge sharing is the most important factor that keeps the team organized to follow an agile procedure. But it is hard to keep regular and proper communication for a team with distributed environment. There are various communication platforms available online to enables this. One of such platforms that we used to keep us close and engaged is discord. It is a free professional chatting app where we can text, call and share files. We can create channels for a group of people, which is called as discord servers through which shared our ideas and knowledge’s by means of messages, voice calls and screen sharing. We used to pin the important messages which we have to keep on track/remind. These pinned messages can be directly accessed by viewing them in pinned messages list. It also has search options through which we can search for some older messages instead of scrolling for a long time. We used discord to create separate channel for discussions and organize daily stand-ups and sprint meetings.

Lucid chart: It is a web-based application for drawing, designing, and sharing charts and diagrams in a collaborative manner across users. It is platform that could be used to visualize our data, architectures, and plan. We used lucid chart create architectures, use case diagrams, framework designs and flow diagram for our project, which helped in visualizing our project and helping us to plan project execution.

Soap UI: We have used REST (Representational State Transfer) API for having communication between the front-end servers to get inputs from users and processing it in backend. To test the REST Api we used soapui. The soap ui is an open-source desktop application designed specifically to inspect, test, invoking, simulating and functional testing of web services using API. It is a cross platform application and supports most of all the standard protocols and technologies. It is best known for its user-friendly client interface which helps new users to learn soon. We used soap ui for regression testing and integration testing. We did not implement automation testing as our project is small which can be tested manually.

Google Docs: Google docs is a web based unified office suite for creating, editing, and maintaining documents online. These documents can be accessed just by using a web browser with an internet access from anywhere around the world. We can also import or export the documents from/to local PC for offline purpose. Even though our team are working from different time zones, this application has made it easy to manage documents with ease of access without depending on one another. We used this application for creating this project report, taking note of important points (like test cases, story points, project changes, etc,.) and editing them parallel in document during our discussions in discord, creating presentation, etc,. The complete application is provided for free of cost by Google.

5.SPRINT

Our product requirement was broken down into many user stories, which formed the product backlog. As seen below, it consists of a variety of tasks and features:

Story title: Create initial design mock-ups for project
Story: As a team I want to know what our website should look like so, everyone is informed on the future view of the project
Tasks: -create mock designs for the website -talk about user interface and improvements -pick the final design that team agrees with
Acceptance criteria: everyone in the team agrees on a design mock-up

Story title: build a Landing Page
Story: As a user I want to view the website. so, I can get info about Covid
Tasks: -write code -Design template -create the test scenario
Acceptance criteria: A user can view the landing page

Story title: Signup and login page
Story: As a user I want to login to the website. so, I can book or view my covid test results
Tasks: -write code -Design page -review the tests with the team -create the test environment -perform the test cases
Acceptance criteria: A user can login to the website

Story title: Forgot password & reset password page
Story: As a user I want to recover or reset my password. so, I can login to the website
Tasks: -write code -Design page -set up the development environment
Acceptance criteria: A user can retrieve lost password to the website

Story title: Dashboard/Home Page
Story: As a user I want to view the home page. so, I can view all the functionality the website can provide
Tasks: -write code -Design page -create the test scenario -review the tests with the team -create the test environment -perform the test cases -report the bugs -code review and bug fixing -set up the development environment
Acceptance criteria: A user can view the dashboard

Story title: Covid Global Data set collection
Story: As a user I want to view the covid statistics. so, I can stay informed
Tasks: -write code -Design page -create the test scenario -review the tests with the team -create the test environment -perform the test cases -report the bugs -code review and bug fixing -set up the development environment
Acceptance criteria: A user can view the covid statistics

Story title: Web Scraping data sets for covid based on users from well-known websites
Story: As a user I want to view the covid statistics from different websites. so, I can stay informed
Tasks: -write code -Design page -create the test scenario -review the tests with the team -create the test environment -perform the test cases -report the bugs -code review and bug fixing -set up the development environment
Acceptance criteria: A user can view the covid statistics form various websites

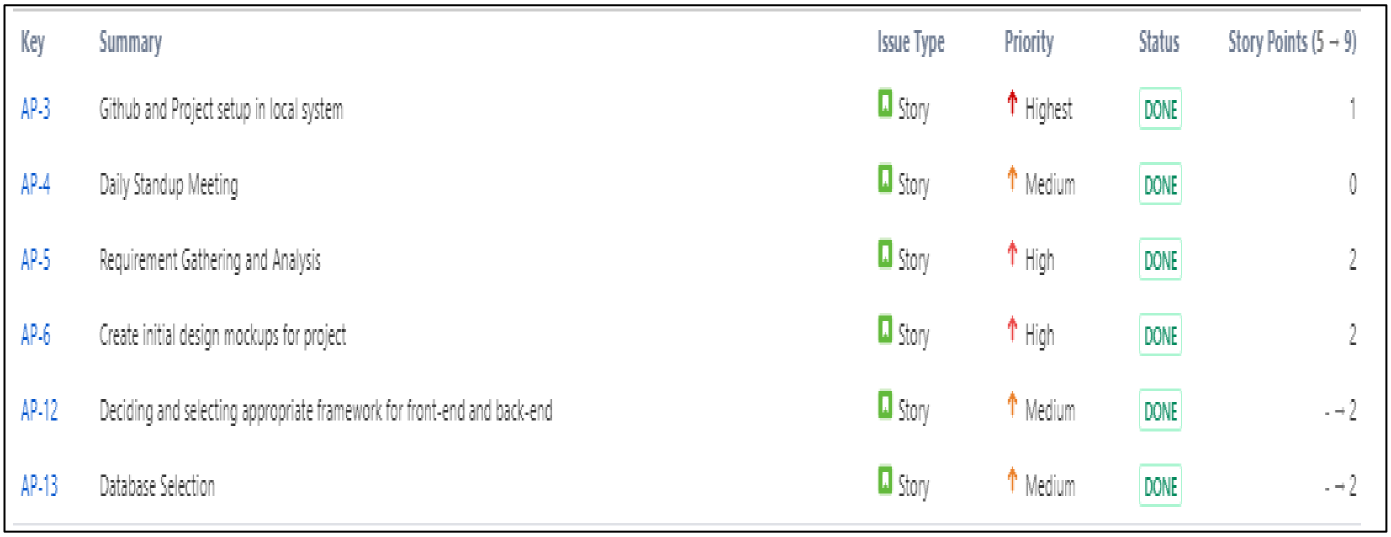
5.1 Sprint 1

Objective: To learn tools and software and analyze the framework and tools for starting the development. To ensure whether all necessary data sets for development are collected.

Backlogs:

- To gather requirements for the project to be started and analyze the use case to determine the necessary data sets to be obtained for the use during the development of application.
- Initializing github accounts and organize the access to the repository for smooth flow of code integrations during development. Also decide on the necessities for deployment of the project in the local system.
- Using and analysing the frameworks that can be used for front end and back end. Based on the analysis of different frameworks and the knowledge among the team members need to decide the appropriate framework and deploy.
- Deciding the suitable database that can be used based on the framework and usage of the product. The database is then to be studied and designed for integration.

- To create a mock design for the application and structure them to have overview of the application to provide a basic knowledge of the project structure and implementation.
- Need to organize team for regular daily scrums/stand up meetings.



Key	Summary	Issue Type	Priority	Status	Story Points (5 → 9)
AP-3	Github and Project setup in local system	Story	↑ Highest	DONE	1
AP-4	Daily Standup Meeting	Story	↑ Medium	DONE	0
AP-5	Requirement Gathering and Analysis	Story	↑ High	DONE	2
AP-6	Create initial design mockups for project	Story	↑ High	DONE	2
AP-12	Deciding and selecting appropriate framework for front-end and back-end	Story	↑ Medium	DONE	- → 2
AP-13	Database Selection	Story	↑ Medium	DONE	- → 2

Figure 17 PRODUCT BACKLOG

The above figure represents the backlogs included in the sprint 1. Using JIRA each backlog is defined with story points based on the workload involved in it. And the priorities of each item are defined based on which the team works to get an expected result and avoid future confusions.

Sprint Review Meeting:

Status:

- Prateek Asthana:
  - Organize team to have scrum / stand up meetings. [DONE]
  - Reviewed all sprint backlogs. [DONE]

- Jun and Vedang:
  - Studied and tested the usage of different frameworks. [DONE]
  - Analyzed the best framework for front and back-end development. [DONE]
  - Gathered the knowledge level about framework other team members and their acceptance for decided framework. [DONE]

- Srikanth and Malavi:
  - Gathered the requirements based on the product. [DONE]
  - Analyzed the use case and determine the datasets needed for development of application. [DONE]
  - Collect the necessary datasets sources for the project. [DONE]

- Mohan Ram:
  - Based on framework and requirements of the product, determine the usage of database. [DONE]
  - Based on usage, analyse the most appropriate database that can be integrated into the framework. [DONE]

- Kolin and Sharukh:
  - Design the use case diagram of the project to have an overview. [DONE]
  - Design the basic mock web pages format of the application to have visualization for better understanding of application. [DONE]

- Vedang and Shravan:
  - Create Git hub account and repositories for the project. [DONE]
  - Manage and share the access for repository and provide KT for the team members. [DONE]
  - Deploy the project system locally and initiate the project system locally. [DONE]

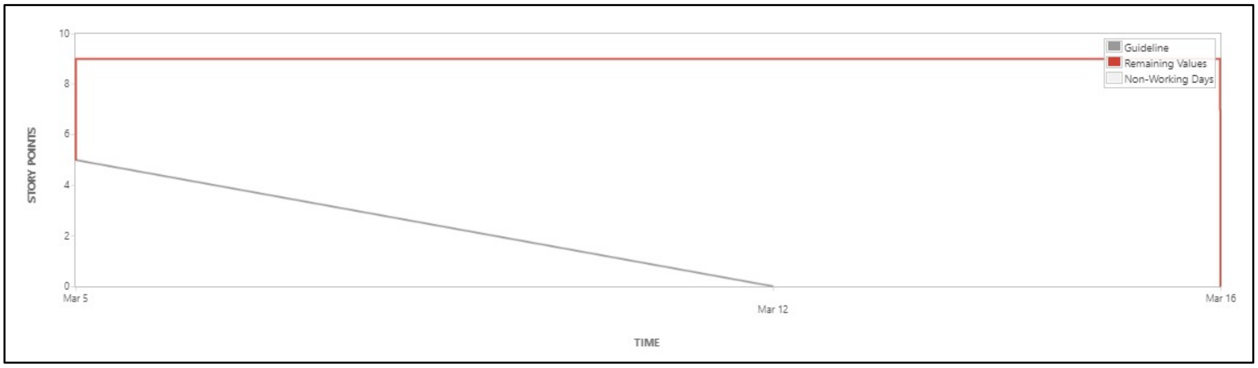


Figure 18 BURNDOWN CHART SPRINT 1

Remarks /Required Actions:

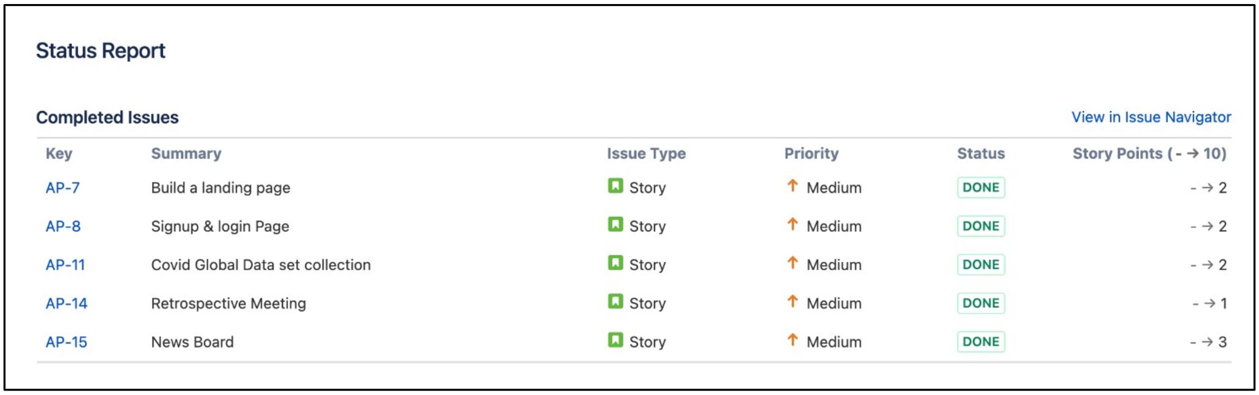
- Different time zones of team members and their availability have reduced the length of daily stand-ups. This was reduced by obtaining a common time that suits all the members and organizing them.
- Though all the members of the team received knowledge transfer on JIRA, lack of previous work on it resulted in blunder while updating. This has been corrected by seeking help from other team members to update JIRA initially.

5.2 Sprint 2

Objective: To empathize the thoughts of the end-users and make an easy-going and efficient website prototype to have a clear look of the page. Besides that, group members who were testers gave us multiple thoughts and helped to make a user interface.

BACKLOGS

- Analyze different people’s emotions and integrated it into a user experience. Created a flow and the integration for the backend and the front end.
- Interaction buttons were made with slight motions to keep the webpage up to date with the recent trends.
- Added texts and emphasized titles where necessary.
- Picked a base color to give the user’s eye a smooth view and feel warmth.
- A mock design was created by number of discussions with the team members.
- Need to organize team for regular daily scrums/stand up meetings.
- Statistics and a nearby location option was created in the mock-up.



Status Report

Completed Issues

View in Issue Navigator

Key	Summary	Issue Type	Priority	Status	Story Points (- → 10)
AP-7	Build a landing page	Story	↑ Medium	DONE	- → 2
AP-8	Signup & login Page	Story	↑ Medium	DONE	- → 2
AP-11	Covid Global Data set collection	Story	↑ Medium	DONE	- → 2
AP-14	Retrospective Meeting	Story	↑ Medium	DONE	- → 1
AP-15	News Board	Story	↑ Medium	DONE	- → 3

Figure 19 PRODUCT BACKLOG

Sprint Review Meeting:

Status:

- Prateek Asthana:
  - Organize team to have scrum / stand up meetings. [DONE]
  - Reviewed all sprint backlogs. [DONE]

- Jun and Vedang:
  - Studied the design to see if all features can be implemented. [DONE]
  - Analyzed the integration for front and back-end development. [DONE]
  - Gathered the knowledge level about the design and started implementing it to the development phase. [DONE]

- Srikanth and Malavi:
  - Gathered the requirements based on the product. [DONE]
  - Tested the Ux designs if the typography, interactions, color were well-adjusted. [DONE]

    - Collect the necessary datasets sources for the project. [DONE]

- Mohan Ram:
    - Based on framework and requirements of the product, determine the usage of database for the second time. [DONE]
    - Based on usage, analyse the most appropriate database that can be integrated into the framework. [DONE]

- Kolin and Sharukh:
    - Design the use case diagram of the project to have a overview. [DONE]
    - Finally created the protype which could be interacted. [DONE]

- Vedang and Shravan:
    - Pushed a lot of the development phases from the prototyping phases in the GitHub repository. [DONE]
    - Manage and share the access for repository and provide KT for the team members. [DONE]
    - Deploy the project system locally and initiate the project system locally again. [DONE]

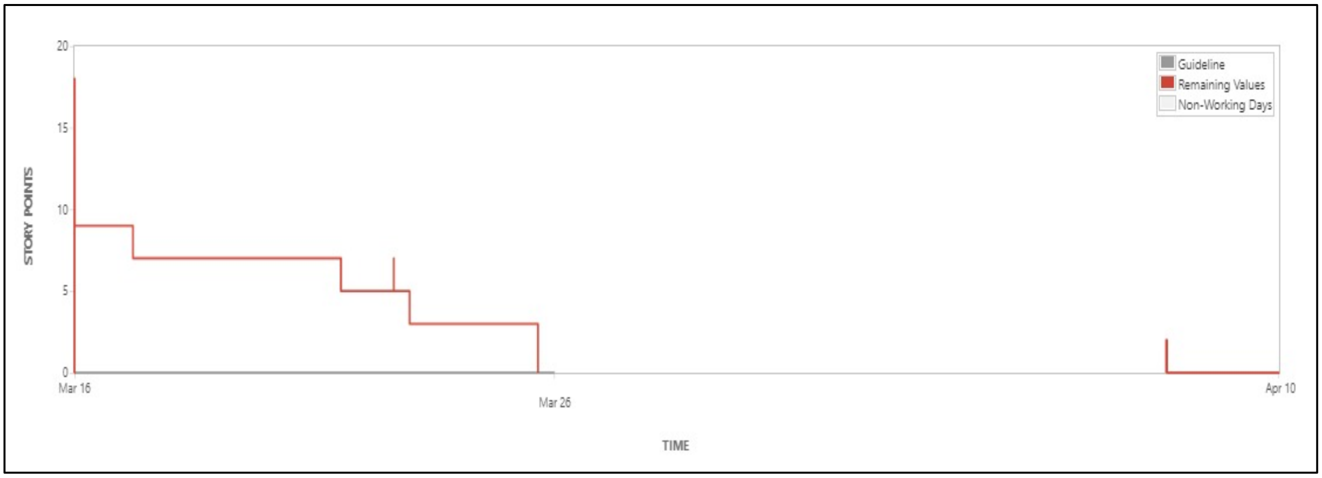


Figure 20 BURNDOWN CHART SPRINT 2

Remarks /Required Actions:

- Different time zones of team members and their availability have reduced the length of daily stand-ups. This was reduced by obtaining a common time that suits all the members and organizing them.
- A prototype was created using Figma.

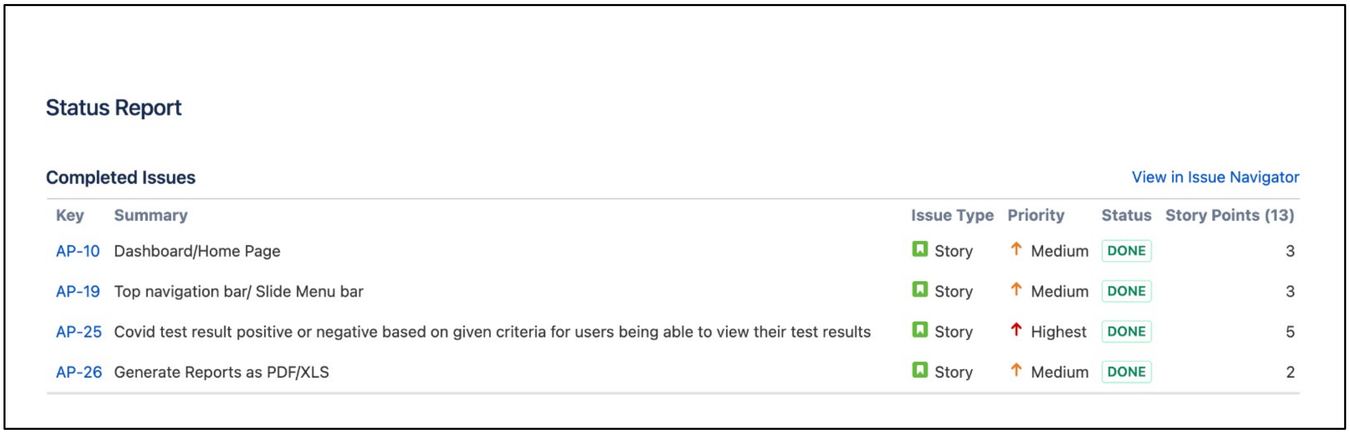
5.3 SPRINT 3

Objective: It was our last sprint and we needed to wrap up everything and include all the important stories into the sprint that will bring the most value from the product.

BACKLOGS

We have included stories worth of 13 story points which covers important features of web application such as dashboard page, top navigation bar, and ability to view covid test result and list of all patients.
Major Task includes:

- Create new react component to develop navigation bar as designed.
- Merge all the task branches and push all the changes to develop branch in GitHub.
- Create new API end points in backend for getting list of patients and covid result.
- Insert and fetch data from cloud MongoDB.
- Connect frontend to backend APIs to fetch data.
- Generate report based on sprints finished.



Status Report

Completed Issues

Key	Summary	Issue Type	Priority	Status	Story Points (13)
AP-10	Dashboard/Home Page	Story	↑ Medium	DONE	3
AP-19	Top navigation bar/ Slide Menu bar	Story	↑ Medium	DONE	3
AP-25	Covid test result positive or negative based on given criteria for users being able to view their test results	Story	↑ Highest	DONE	5
AP-26	Generate Reports as PDF/XLS	Story	↑ Medium	DONE	2

Figure 21 PRODUCT BACKLOG

Sprint Review Meeting:

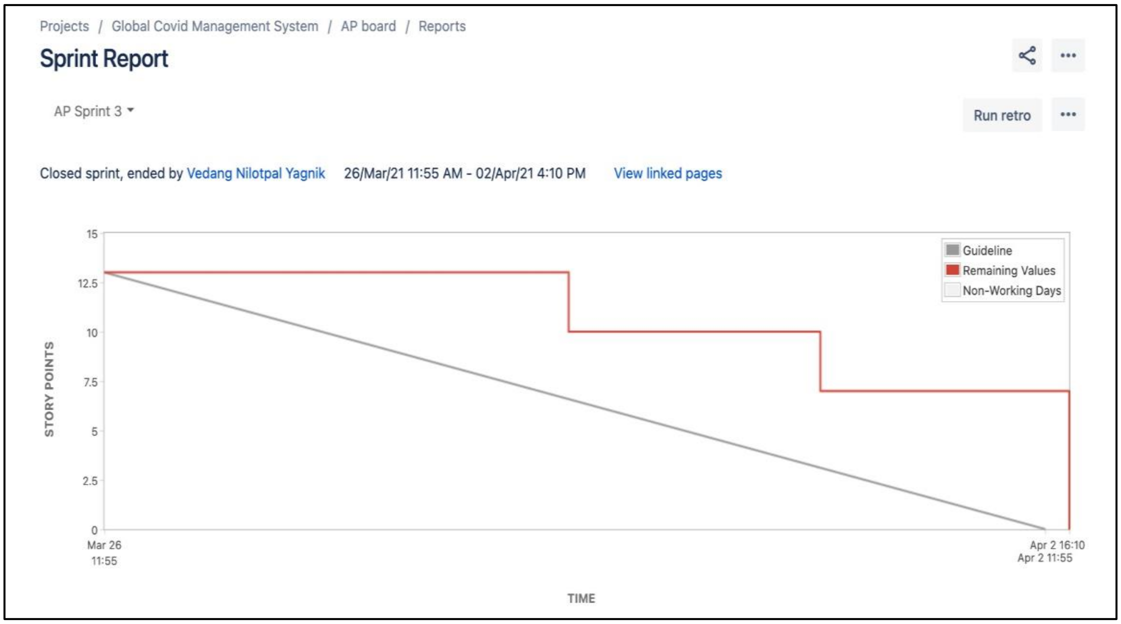
Story Image:

We have included stories worth of 13 story points which covers important features of web application such as dashboard page, top navigation bar, and ability to view covid test result and list of all patients.

- Prateek Asthana:
  - Organize team to have scrum / stand up meetings. [DONE]
  - Reviewed all sprint backlogs. [DONE]
- Jun and Vedang:

- Connect frontend to backend APIs to fetch data. [DONE]
- Insert and fetch data from cloud MongoDB. [DONE]

- Srikanth and Malavi:
  - Generate report based on Sprints accomplished. [DONE]
  - Final testing of the demo product. [DONE]
- Mohan Ram:
  - Merge all the task branches and push all the changes to develop branch in GitHub. [DONE]
- Kolin and Sharukh:
  - Create new API end points in backend for getting list of patients and covid result [DONE]
- Vedang and Shravan:
  - Create new react component to develop navigation bar as designed. [DONE]


Projects / Global Covid Management System / AP board / Reports

Sprint Report

AP Sprint 3

Closed sprint, ended by Vedang Nilotpal Yagnik 26/Mar/21 11:55 AM - 02/Apr/21 4:10 PM View linked pages

Figure 22 BURNDOWN CHART

6. IMPEDIMENTS

- We worked in a Distributed Agile model, people working in this project are of different backgrounds, different languages. So, we had this impediment on the starting phases of the project.

- It was difficult for all to schedule a time which was convenient to all as people were in different parts of world due to remote learning.
- Maintaining Jira was difficult as we were learning Latest. New languages and new tools for web development means the learning curve is there to adapt to it. So, it took time for developers to enhance their knowledge.
- Delay in accessing GitHub and server DB.
- COVID 19 Pandemic
- Integration Issues

7. RESULTS AND DISCUSSION

The software industry has yet to develop a covid management framework, and we plan to do so. It will include features that allow users to build profiles that include their name, address, phone number, job title, current earnings, and people in touch, as well as the ability for health care professionals to input covid results into the system, which the public will access. We will use that information to build geo data analytical maps that display covid hotspots, job losses due to covid, and current population earnings by area. For any outbreak, multiple databases will have to be linked across sectors to include information from people with and without the disease. This kind of covid development is useful all over the world to categorize people in between whether they are covid-19 positive or not and later use that saved data for prediction and analysis in the future using different machine learning techniques. For any outbreak, multiple databases will have to be linked across sectors to include information from people with and without the disease. This kind of covid development is useful all over the world to categorize people in between whether they are covid-19 positive or not and later use that saved data for prediction and analysis in the future using different machine learning techniques.

This kind of covid people saved data will be useful in future to determine what kind of characteristics people find in covid-19 positive people.

8. CONCLUSION

The agile methodological approach to software development and application design has recently became one of the most employed methods in the industry. Our approach to product development was derived from the 12 principles of Agile (Agile Manifesto, 2001). We developed a global covid management system using agile SCRUM methodology in which we divided the product into multiple user stories and worked upon them. It involved frequent development, incessant refinement at every phase and continuous communication with the team members to ensure a valued product delivery. In agile, we mainly focused on Product implementation over detailed documentation and reacting to transition despite a strategic plan. This ensured us to deliver a final software product by frequent refining in accordance with the changing needs of customer and met the requirements in the end. We used scrum framework for our agile approach. Scrum is iteration planning, iteration execution and delivering the result continuously. Despite facing many impediments in the process, we ensured good team collaboration and tackled them effectively.

Frequent communication with the team in form of daily stand-up meetings, sprint planning, retrospective meetings contributed for the success as a team and helped us a lot in delivering a fruitful product in the end.